

程序员代码面试指南

IT名企算法与数据结构题目最优解

左程云 著

精选IT名企真实代码面试题，全面覆盖算法与数据结构题型



Coding Interview Guide

程序员代码面试指南

IT名企算法与数据结构题目最优解



左程云 著

电子工业出版社
Publishing House of Electronics Industry

内 容 简 介

这是一本程序员面试宝典！书中对IT名企代码面试各类题目的最优解进行了总结，并提供了相关代码实现。针对当前程序员面试缺乏权威题目汇总这一痛点，本书选取将近200道真实出现过的经典代码面试题，帮助广大程序员的面试准备做到万无一失。“刷”完本书后，你就是“题王”！

本书采用题目+解答的方式组织内容，并把面试题类型相近或者解法相近的题目尽量放在一起，读者在学习本书时很容易看出面试题解法之间的联系，使知识的学习避免碎片化。书中将所有的面试题从难到易依次分为“将、校、尉、士”四个档次，方便读者有针对性地选择“刷”题。本书所收录的所有面试题都给出了最优解讲解和代码实现，并且提供了一些普通解法和最优解法的运行时间对比，让读者真切地感受到最优解的魅力！

本书中的题目全面且经典，更重要的是，书中收录了大量独家题目和最优解分析，这些内容源自笔者多年来“死磕自己”的深入思考。

码农们，你们做好准备在IT名企的面试中脱颖而出、一举成名了吗？这本书就是你应该拥有的“神兵利器”。当然，对需要提升算法和数据结构等方面能力的程序员而言，本书的价值也是显而易见的。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

程序员代码面试指南：IT名企算法与数据结构题目最优解 / 左程云著. —北京：电子工业出版社，2015.9

ISBN 978-7-121-27011-6

I. ①程… II. ①左… III. ①程序设计—工程技术人员—资格考试—自学参考资料 IV. ①TP311.1

中国版本图书馆 CIP 数据核字(2015)第 198018 号

策划编辑：牛 勇

责任编辑：李利健

印 刷：三河市双峰印刷装订有限公司

装 订：三河市双峰印刷装订有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×980 1/16 印张：33.25 字数：658.9 千字

版 次：2015 年 9 月第 1 版

印 次：2015 年 9 月第 1 次印刷

定 价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

献给左军和谢桂兰

特别说明

1. 本书所有题目的代码都为 Java 实现，但这并不会妨碍其他语言使用者的阅读。这是因为笔者在实现每一道题目时，都尽最大努力回避与 Java 语言特性相关的写法出现，而且尽量遵循大多数编程语言共有的写法习惯。所以，将本书中的 Java 实现改写成其他语言的实现是非常容易的。

2. 在 Java 中，如果想得到字符串 `str` 第 i 个位置的字符，需用如下方式：

```
char p = str.charAt(i);
```

本书提供的函数中有大量参数为字符串类型的函数，但如上所示的方式并不符合大多数读者的阅读习惯。为了让代码更加易读，笔者都在这样的函数中把字符串类型的参数转换成 `char` 类型数组的变量来使用，例如：

```
char[] charArr = str.toCharArray();
```

此时得到字符串 `str` 第 i 个位置的字符，可以用如下方式：

```
char p = charArr[i];
```

在本书中，发生如上转换行为的函数在估算额外空间复杂度的时候，笔者并没有把 `charArr` 的空间计算在内，这是因为如果不转换成 `char` 数组，而是选择直接使用原参数 `str`，也是完全可以的，之所以选择转换，仅仅是为了让读者更容易读懂代码；是否进行转换对算法的逻辑没有任何影响，所以不把 `charArr` 的空间算作必须使用的额外空间。

另外，本书涉及的程序源代码可以在 <http://www.broadview.com.cn/27011> 中下载。

推荐序 1

2015 年春节，因为公司业务的快速发展，我们开始寻觅优秀的笔试面试算法讲师。几经周折，找到了当时在举办线下算法分享的程云，认认真真地听他讲了一堂课，当时就认定他就是我们要找的人。

我听过很多国内顶尖 ACM 选手的算法分享，但是每一次听完以后总觉得我和那些人永远隔着一个断裂带，算法对我来说遥不可及，而程云讲解算法的时候总能从最小的切口讲起，由浅入深，环环相扣，不知不觉引你走向算法的核心精髓，那种醍醐灌顶的感觉能激发大家学习算法的热情，并一直推着我们前进。

这几年 IT 技术蓬勃发展，日新月异，对技术人才的需求日益增长，程序员招聘市场也如火如荼。在有限的三五轮面试中，国外流行让面试者编程解决某些数据结构和算法的题目，通过观察面试者编码的熟练程度、思考的速度和深度来衡量面试者的能力和潜力。国内以百度、阿里、腾讯为首的互联网企业也都逐步开始采用算法面试来筛选人才。

程云出于对算法的热爱，长期泡在 `careercup`、`leetcode` 等笔试面试网站上，编码解决各种最新的笔试面试编程题，对各种笔试面试编程题的解题技巧了如指掌。

算法面试普及后，传统的数据结构和算法课本讲得太过基础，又远离求职需求，国内也逐渐出现迎合求职需求的笔试面试工具书，这些书籍有些过于应试，纯粹以通过面试为导向，程云的书和那些书相比，题目更前沿，讲解更注重思考思路和代码的实践技巧，对每个题目都深挖最优解，同时根据自己在线下讲课学员们的反馈，对每个编程考题的解题反复修改，让思路更清晰。

这本书不仅可以作为面试代码指南，还可以作为学生课后的辅助练习，“刷”题 5 年，悉数总结都沉淀在这本书里，相信读者跟着他的引导从头到尾逐一攻克一定会有所收获。

叶向宇
牛客网 CEO

推荐序 2

初次遇进程云是在 2014 年 8 月，当时我在上一家公司工作刚好满 4 年，也是在那时我开始想换个环境，寻找新机会，就试着投了一家公司，结果第一次面试遇到算法题就被淘汰了。后来又面试过其他一些国内互联网公司，也总是卡在算法上。其实，之前我曾经自己在家抱着《算法导论》“啃”了几章，花了 1 个月的业余时间看了前 5 章，后面就没再继续坚持下去。看过的人都知道，虽然很有用，但实在很难“啃”。

单调地看书很枯燥，于是想到去网上找志同道合的人一起研究，就开始“逛”算法论坛。很巧的是，在某个论坛的算法板块看到一个帖子，说是在周末有算法交流班，当时我立即报名，周日的名额已满，我是很幸运地“替补”上去的。

还记得第一次交流是在程云租的房子里，小小的客厅里放了一张沙发、两排椅子和一张桌子，桌上放着笔记本电脑和一台大电视，前面还挂着白板。第一次算法交流就在这样的环境里开始了。

程云讲起话来犹如行云流水，我们听得更是酣畅淋漓，第一次听完就爱上了……当然，我说的是他的讲述。

相信大家都有过这样的经历，面对一道算法题，苦思冥想了半天，还是不知道怎么解，感觉很沮丧。如果这时突然有人把解题思路和方法以及代码都告诉你了，是不是感觉豁然开朗，心情舒畅了？这样的情景一天出现一次就可以让人感觉很开心，而如果一天连续出现二十次，那将会是什么感觉？一个字：爽！

程云把每一道题都讲解得清晰透彻，有的题目难以理解、思路诡异，他就会不厌其烦地反复讲解，用形象的方式展现复杂的逻辑，直到大家都听懂为止。给人的感觉可以说是高潮迭起，一波又一波。

后来进行第二次交流时，我带来最好的朋友一起参加。之后的交流中，我和朋友都毫不犹豫地报名参加。交流的内容涉及经典算法的高难度题目，也有一些小巧玲珑的技巧题。难题难得让人叹服，巧题巧得让人玩味。

对想去国外大公司就职的程序员来说，算法题这一关是必不可少的。程云讲述的题目

是他 5 年“刷”题的经验积累而成的，其实只要掌握题目的解题思路和思想，就足以应付国内互联网公司程序员职位的算法面试题。不过，要想去国外的大公司，比如 Google、Facebook 之类的，还是要研究得透彻一些才行。

另外，除应付面试之外，还有很重要的一点，甚至是更重要的一点，就是本书可以帮助我们打开思路，因为很多算法题的解法是需要逆向思维的，需要跳出原有的固定思维模式，当思维模式被打开之后，你会发现原有的事物现在看起来会有不同的看法，因为角度变了。不过这只能自己体会。

后来才知道，程云举办算法交流是为写书做准备。用他的话说：“会做题不算什么，比我“刷”题多的人我也能找出一大堆，但能给人讲明白就不容易了。”于是我后来又变成了程云在写这本书期间的试读者。

在此书还未上市之前，就能听到作者面对面地逐一讲解每一道题，真是非常难得且宝贵的经历。

如果你和我一样，对数据结构有个大概的了解，很想快速掌握算法题的解法技巧，那么这本书一定适合你！

祝每一位勤奋努力的程序员都能拿到自己满意的职位！

周宝鑫
一个程序员

自序

我能出书挺意外的。

在 6 年前的某一天，虽然我早就知道想进入那些大公司要靠“刷”代码面试题来练习编写代码的能力。可是这一天却不止如此，我突然有了心情去看代码面试题长什么样子，于是收集了代码面试的题目，越深入，我越有一种恐慌的感觉，因为感觉自己什么都不太在行，对一个归并排序（Merge sort）写出完整的代码都感觉挺费劲的，面对这个冯·诺伊曼发明的排序算法，我真有底气说自己是计算机专业的学生吗？这种打击并没有持续太久，因为爱耍小聪明的人总会特别自信。我决定开始认真面对“刷”题这件事，但那时我根本不知道我即将面对什么，更不要谈有写书的念头。

我把课余时间利用起来，心想：不就是“刷”题吗？别人能写出来，咱也能写出来。起初的心态是我不服，我就想告诉自己能行。过程虐心是肯定的，经常半夜因为看到一个复杂度特别低的算法自己真的不能理解而沮丧地睡不着觉。当时觉得找不到什么资料能彻底让我明白，书上讲得太粗浅，网上的太散乱，代码写得看不懂。起初我“刷”题的时候无数次地想放弃，因为觉得这些都是什么玩意儿！我为什么放着好好的日子不过，去找这种罪受？可是我又不甘心，虽然我不懂很多解法，但是它们真的很有意思。

我将能买到的所有相关书籍上的所有题目全都研究了一遍，不管是中文的还是英文的，我都硬着头皮“啃”。写完每道题后，我都和书上的方法进行反复对比。“啃”完了五六本书之后，距离我刚开始“刷”题已经过去 16 个月了。写书？别逗了，才刚看完。

“年轻人总会找借口说这个东西不是我感兴趣的，所以我做不好是应该的。但他们没有注意的是，你面对的事情中感兴趣的事情总是少数，这就使得大多数时候你做事情的态度总是很懈怠、很消极，这使你变成了一个懈怠的人。当你真正面对自己感兴趣的東西时，你发现你已经攥不紧拳头了。”时常想起本科时的毕业设计指导老师——高鹏义老师说的这句话。说得对！对一个东西，如果你没有透彻研究过，不要轻易说它不精彩。这不是博爱，而是对自己认真。

“刷”题代码达到 4 万行的时候，我基本上成了国内外所有热门“刷”题网站的日常用户，此时我确认了一件事情，今天的代码面试指导真的处在一个很初级的阶段，这种不健全是

全方面的。

例如：

- 经常看到一篇文章前后的语境是割裂的，作者经常根据之前的一个优良解法提出更好的优化方式，但整篇文章都不提之前的解法是什么。这就导致初学者根本无法看懂；
- 几乎所有的书籍都忽略例子带来的引导作用，甚至还有不少书籍在阐述一个解法的时候只写伪代码，这就使得读者在看懂意思和自己真正能写出代码之间其实还有很长的路要走；
- 代码面试题的特点是“多”、“杂”、“难”，从着手开始学习到最终达到自己想要的效果之间，自己对自己的评估根本无从谈起。“慢慢练吧，学海无涯”成为主要的心态，这就难免会产生怀疑的情绪；
- 看见一道新的面试题时还是会无从下手，因为之前的学习无法做到举一反三，对自己做过的题目缺乏总结和归纳。

难道“刷”题真的只适合聪明人玩？我不这么看，既然大多数内容处在有待商榷的地步，那我就去学习原论文吧。

当时一个人在国外，记得在初冬的一个下午，“刷”题已经两年之久，快吃晚饭的时候，我突然想起自己忘了吃午饭，就冲出家门去觅食。站在 7-11 门前的广场上，我拿着 1.5 美元的热狗和 75 美分的咖啡，微温的阳光撒在眼睛里，远远地望着即将消失的一天。我停下来，把咖啡放在斑驳的石头台子上，手里的热狗挺好看，香肠和洋葱都挺新鲜，清冷的空气吹过来，却让我的心绪更乱。旧金山的天空五彩斑斓，让漂泊者头晕目眩。哭得跟个鬼似的我除了想家，哪里敢想自己会出书呢？

当我意识到在网上很难搜到新鲜的题目时，我已经换了两家公司，反复实现了 600 多道题目，写了差不多 10 万行代码。原来只是为了找份工作而“刷”题这一初心早就忘了，变成了兴趣并坚持了这么久，我自己也感到意外。更奇怪的是，我已经完全乐在其中，同时交流欲望越来越强，时常和同事们展开这方面的讨论。发现很多书上的解法不是最优，很多题目其实和同事们讨论的做法更好，发现高手特别多，但好像都懒得动笔。

有一天，我看到自己写的题目，想到自己那些抓心挠肝的日子，突然觉得要不出书吧？我已经离不开这种感觉了，如果这不是真爱，那什么才是呢？

这不是一个励志的故事，是一个爱“刷”题的人决定把很多最优解讲出来，就这么简单。

左程云

2015 年 7 月 20 日

目 录

第 1 章 栈和队列.....	1
设计一个有 getMin 功能的栈 (士 ★☆☆☆)	1
由两个栈组成的队列 (尉 ★★☆☆)	5
如何仅用递归函数和栈操作逆序一个栈 (尉 ★★☆☆)	8
猫狗队列 (士 ★☆☆☆)	10
用一个栈实现另一个栈的排序 (士 ★☆☆☆)	13
用栈来求解汉诺塔问题 (校 ★★★☆)	14
生成窗口最大值数组 (尉 ★★☆☆)	19
构造数组的 MaxTree (校 ★★★☆)	22
求最大子矩阵的大小 (校 ★★★☆)	26
最大值减去最小值小于或等于 num 的子数组数量 (校 ★★★☆)	31
第 2 章 链表问题.....	34
打印两个有序链表的公共部分 (士 ★☆☆☆)	34
在单链表和双链表中删除倒数第 K 个节点 (士 ★☆☆☆)	35
删除链表的中间节点和 a/b 处的节点 (士 ★☆☆☆)	38
反转单向和双向链表 (士 ★☆☆☆)	40
反转部分单向链表 (士 ★☆☆☆)	42
环形单链表的约瑟夫问题 (原问题: 士 ★☆☆☆ 进阶: 校 ★★★☆)	43
判断一个链表是否为回文结构 (普通解法 士 ★☆☆☆) (进阶解法 尉 ★★☆☆)	48
将单向链表按某值划分成左边小、中间相等、右边大的形式 (尉 ★★☆☆)	52
复制含有随机指针节点的链表 (尉 ★★☆☆)	56
两个单链表生成相加链表 (士 ★☆☆☆)	59

两个单链表相交的一系列问题 (将 ★★★★★)	62
将单链表的每 K 个节点之间逆序 (尉 ★★☆☆)	68
删除无序单链表中值重复出现的节点 (士 ★☆☆☆)	71
在单链表中删除指定值的节点 (士 ★☆☆☆)	73
将搜索二叉树转换成双向链表 (尉 ★★☆☆)	74
单链表的选择排序 (士 ★☆☆☆)	79
一种怪异的节点删除方式 (士 ★☆☆☆)	81
向有序的环形单链表中插入新节点 (士 ★☆☆☆)	82
合并两个有序的单链表 (士 ★☆☆☆)	84
按照左右半区的方式重新组合单链表 (士 ★☆☆☆)	86
第 3 章 二叉树问题	88
分别用递归和非递归方式实现二叉树先序、中序和后序遍历 (校 ★★★★★)	88
打印二叉树的边界节点 (尉 ★★☆☆)	95
如何较为直观地打印二叉树 (尉 ★★☆☆)	100
二叉树的序列化和反序列化 (士 ★☆☆☆)	103
遍历二叉树的神级方法 (将 ★★★★★)	107
在二叉树中找到累加和为指定值的最长路径长度 (尉 ★★☆☆)	115
找到二叉树中的最大搜索二叉子树 (尉 ★★☆☆)	117
找到二叉树中符合搜索二叉树条件的最大拓扑结构 (校 ★★★★★)	119
二叉树的按层打印与 ZigZag 打印 (尉 ★★☆☆)	129
调整搜索二叉树中两个错误的节点 (原问题: 尉 ★★☆☆)	
(进阶问题: 将 ★★★★★)	134
判断 t_1 树是否包含 t_2 树全部的拓扑结构 (士 ★☆☆☆)	140
判断 t_1 树中是否有与 t_2 树拓扑结构完全相同的子树 (校 ★★★★★)	141
判断二叉树是否为平衡二叉树 (士 ★☆☆☆)	144
根据后序数组重建搜索二叉树 (士 ★☆☆☆)	145
判断一棵二叉树是否为搜索二叉树和完全二叉树 (士 ★☆☆☆)	147
通过有序数组生成平衡搜索二叉树 (士 ★☆☆☆)	150
在二叉树中找到一个节点的后继节点 (尉 ★★☆☆)	151
在二叉树中找到两个节点的最近公共祖先 (原问题: 士 ★☆☆☆)	
(进阶问题: 尉 ★★☆☆ 再进阶问题: 校 ★★★★★)	153

Tarjan 算法与并查集解决二叉树节点间最近公共祖先的批量查询问题 (校 ★★★☆)	159
二叉树节点间的最大距离问题 (尉 ★★★☆)	169
先序、中序和后序数组两两结合重构二叉树 (先序与中序结合 士 ☆☆☆)	
(中序与后序结合 士 ☆☆☆ 先序与后序结合 尉 ★★★☆)	171
通过先序和中序数组生成后序数组 (士 ☆☆☆)	174
统计和生成所有不同的二叉树 (尉 ★★★☆)	175
统计完全二叉树的节点数 (尉 ★★★☆)	178
第 4 章 递归和动态规划	181
斐波那契系列问题的递归和动态规划 (将 ★★★★★)	181
矩阵的最小路径和 (尉 ★★★☆)	187
换钱的最少货币数 (尉 ★★★☆)	191
换钱的方法数 (尉 ★★★☆)	196
最长递增子序列 (校 ★★★☆)	202
汉诺塔问题 (校 ★★★☆)	206
最长公共子序列问题 (尉 ★★★☆)	210
最长公共子串问题 (校 ★★★☆)	213
最小编辑代价 (校 ★★★☆)	217
字符串的交错组成 (校 ★★★☆)	220
龙与地下城游戏问题 (尉 ★★★☆)	223
数字字符串转换为字母组合的种数 (尉 ★★★☆)	225
表达式得到期望结果的组成种数 (校 ★★★☆)	228
排成一条线的纸牌博弈问题 (尉 ★★★☆)	233
跳跃游戏 (士 ☆☆☆)	235
数组中的最长连续序列 (尉 ★★★☆)	236
N 皇后问题 (校 ★★★☆)	238
第 5 章 字符串问题	242
判断两个字符串是否互为变形词 (士 ☆☆☆)	242
字符串中数字子串的求和 (士 ☆☆☆)	243
去掉字符串中连续出现 k 个 0 的子串 (士 ☆☆☆)	245
判断两个字符串是否互为旋转词 (士 ☆☆☆)	247

将整数字符串转成整数值 (尉 ★★☆☆)	248
替换字符串中连续出现的指定字符串 (士 ★☆☆☆)	251
字符串的统计字符串 (士 ★☆☆☆)	253
判断字符数组中是否所有的字符都只出现过一次 (按要求 1 实现的方法 士 ★☆☆☆)	
(按要求 2 实现的方法 尉 ★★☆☆)	255
在有序但含有空的数组中查找字符串 (尉 ★★☆☆)	258
字符串的调整与替换 (士 ★☆☆☆)	260
翻转字符串 (士 ★☆☆☆)	262
数组中两个字符串的最小距离 (尉 ★★☆☆)	266
添加最少字符使字符串整体都是回文字符串 (校 ★★★☆)	269
括号字符串的有效性和最长有效长度 (原问题 士 ★☆☆☆)	
(补充问题 尉 ★★☆☆)	273
公式字符串求值 (校 ★★★☆)	276
0 左边必有 1 的二进制字符串数量 (校 ★★★☆)	278
拼接所有字符串产生字典顺序最小的大写字母串 (校 ★★★☆)	281
找到字符串的最长无重复字符串 (尉 ★★☆☆)	284
找到被指的新类型字符 (士 ★☆☆☆)	286
最小包含子串的长度 (校 ★★★☆)	288
回文最少分割数 (尉 ★★☆☆)	292
字符串匹配问题 (校 ★★★☆)	294
字典树 (前缀树) 的实现 (尉 ★★☆☆)	299
第 6 章 大数据和空间限制	303
认识布隆过滤器 (尉 ★★☆☆)	303
只用 2GB 内存在 20 亿个整数中找到出现次数最多的数 (士 ★☆☆☆)	308
40 亿个非负整数中找到没出现的数 (尉 ★★☆☆)	309
找到 100 亿个 URL 中重复的 URL 以及搜索词汇的 top K 问题 (士 ★☆☆☆)	311
40 亿个非负整数中找到出现两次的数和所有数的中位数 (尉 ★★☆☆)	312
一致性哈希算法的基本原理 (尉 ★★☆☆)	313
第 7 章 位运算	317
不用额外变量交换两个整数的值 (士 ★☆☆☆)	317

不用任何比较判断找出两个数中较大的数 (校 ★★★☆)	318
只用位运算不用算术运算实现整数的加减乘除运算 (尉 ★★☆☆)	319
整数的二进制表达中有多少个 1 (尉 ★★☆☆)	325
在其他数都出现偶数次的数组中找到出现奇数次的数 (尉 ★★☆☆)	327
在其他数都出现 k 次的数组中找到只出现一次的数 (尉 ★★☆☆)	329
第 8 章 数组和矩阵问题	331
转圈打印矩阵 (士 ★☆☆☆)	331
将正方形矩阵顺时针转动 90° (士 ★☆☆☆)	333
“之”字形打印矩阵 (士 ★☆☆☆)	335
找到无序数组中最小的 k 个数 ($O(N\log k)$ 的方法 尉 ★★☆☆)	
($O(N)$ 的方法 将 ★★★)	336
需要排序的最短子数组长度 (士 ★☆☆☆)	342
在数组中找到出现次数大于 N/K 的数 (校 ★★★☆)	343
在行列都排好序的矩阵中找数 (士 ★☆☆☆)	347
最长的可整合子数组的长度 (尉 ★★☆☆)	349
不重复打印排序数组中相加和为给定值的所有二元组和三元组 (尉 ★★☆☆)	351
未排序正数数组中累加和为给定值的最长子数组长度 (尉 ★★☆☆)	354
未排序数组中累加和为给定值的最长子数组系列问题 (尉 ★★☆☆)	355
未排序数组中累加和小于或等于给定值的最长子数组长度 (校 ★★★☆)	358
计算数组的小和 (校 ★★★☆)	361
自然数数组的排序 (士 ★☆☆☆)	364
奇数下标都是奇数或者偶数下标都是偶数 (士 ★☆☆☆)	366
子数组的最大累加和问题 (士 ★☆☆☆)	367
子矩阵的最大累加和问题 (尉 ★★☆☆)	368
在数组中找到一个局部最小的位置 (尉 ★★☆☆)	371
数组中子数组的最大累乘积 (尉 ★★☆☆)	373
打印 N 个数组整体最大的 Top K (尉 ★★☆☆)	374
边界都是 1 的最大正方形大小 (尉 ★★☆☆)	377
不包含本位置值的累乘数组 (士 ★☆☆☆)	380
数组的 partition 调整 (士 ★☆☆☆)	382
求最短通路值 (尉 ★★☆☆)	384

数组中未出现的最小正整数 (尉 ★★★☆☆)	386
数组排序之后相邻数的最大差值 (尉 ★★★☆☆)	388
第 9 章 其他题目	390
从 5 随机到 7 随机及其扩展 (原问题 尉 ★★★☆☆ 补充问题 尉 ★★★☆☆)	
(进阶问题 校 ★★★☆☆)	390
一行代码求两个数的最大公约数 (士 ★★★☆☆)	394
有关阶乘的两个问题 (原问题 尉 ★★★☆☆ 进阶问题 校 ★★★☆☆)	395
判断一个点是否在矩形内部 (尉 ★★★☆☆)	398
判断一个点是否在三角形内部 (尉 ★★★☆☆)	399
折纸问题 (尉 ★★★☆☆)	402
蓄水池算法 (尉 ★★★☆☆)	404
设计有 setAll 功能的哈希表 (士 ★☆☆☆☆)	406
最大的 leftMax 与 rightMax 之差的绝对值 (校 ★★★☆☆)	408
设计可以变更的缓存结构 (尉 ★★★☆☆)	410
设计 RandomPool 结构 (尉 ★★★☆☆)	414
调整 $[0,x]$ 区间上的数出现的概率 (士 ★☆☆☆☆)	416
路径数组变为统计数组 (校 ★★★☆☆)	417
正数数组的最小不可组成和 (尉 ★★★☆☆)	422
一种字符串和数字的对应关系 (校 ★★★☆☆)	426
1 到 n 中 1 出现的次数 (校 ★★★☆☆)	429
从 N 个数中等概率打印 M 个数 (士 ★☆☆☆☆)	431
判断一个数是否是回文数 (士 ★☆☆☆☆)	433
在有序旋转数组中找到最小值 (尉 ★★★☆☆)	434
在有序旋转数组中找到一个数 (尉 ★★★☆☆)	436
数字的英文表达和中文表达 (校 ★★★☆☆)	439
分糖果问题 (校 ★★★☆☆)	444
一种消息接收并打印的结构设计 (尉 ★★★☆☆)	448
设计一个没有扩容负担的堆结构 (将 ★★★★★)	451
随时找到数据流的中位数 (将 ★★★★★)	462
在两个长度相等的排序数组中找到上中位数 (尉 ★★★☆☆)	465
在两个排序数组中找到第 K 小的数 (将 ★★★★★)	468
两个有序数组间相加和的 TOP K 问题 (尉 ★★★☆☆)	471

出现次数的 TOP K 问题 (原问题 尉 ★★☆☆ 进阶问题 校 ★★☆☆)	474
Manacher 算法 (将 ★★★★★)	483
KMP 算法 (将 ★★★★★)	491
丢棋子问题 (校 ★★☆☆)	498
画匠问题 (校 ★★☆☆)	505
邮局选址问题 (校 ★★☆☆)	509